Confident Association for Long-term Tracking

Zhaohui Zuo
College of Control Science and Engineering, Beijing University of Chemical Technology, Beijing, China

Abstract—Aiming at the exponential growth of solution scale in multiple hypothesis tracking (MHT), a continuous consistency model (CCM) is proposed. The key to improve MHT performance is to improve the efficiency of branch management. However, due to the inevitable detector failure, when the tree is expanded and each detection is organized as the root node of the new tree, a large number of virtual nodes are used. This leads to rapid growth of branches. Different from previous MHT implementations, CCM divides detection into four categories, including continuous, left continuous, right continuous and discontinuous. Comparative experiments show that CCM has significantly improved the computational efficiency and obtained the most advanced results on MOT challenge benchmark.

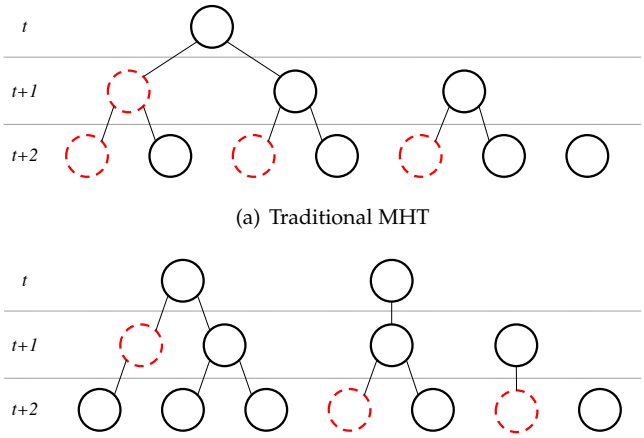
Index Terms—Visual Tracking, Multiple Hypothesis Tracking, Data Association

1 Introduction

Tracking multiple targets has been an important topic in the field of computer vision. Although significant progress has been made, there are still many tough unsolved problems. Similar appearances, frequent occlusions and motion blur (camera shakes), for instance, are some common obstacles for tracking.

Multiple hypothesis tracking (MHT) [1] is one of the most successful frameworks to solve these problems. It keeps trees of hypotheses for targets and evaluates the likelihood of each branch to select the most likely one. However, exponential growth of the solution space is the critical defect of MHT. The scale and computational complexity grow dramatically along with the number of frames and detections. There are some strict but compromising constraints to control its growth, such as the number of the hypothesis trees, the number of the leaf nodes, the maximal number of dummy nodes. In addition, an iterative updating technique [2] is applied for speeding up, but it does not address the fundamental issue of the growth.

A main reason of the rapid growth is the strategy that every leaf node is extended with a dummy node, and every detection is built as the root node of a new tree, as shown in Figure 1(a). In this paper, we propose a novel continuous consistency model (CCM) to characterize the relationship between targets in adjacent frames. CCM categorizes detections into four typical types including continuous, left continuous, right continuous and discontinuous. As shown in Figure 1(b), it reduces the scale of tracking trees by controlling hypothesis generation process. In addition, we remove the impractical constraints on dummy nodes by exploiting the CCM. As a result, our method significantly improves the computational efficiency while achieving better tracking performance. Comparative experiment results show our method is effective in restraining the exponential growth of MHT and has strong ability to reduce the risk of identity switch caused by long-term occlusions. The contributions of this paper are:



(a) Traditional MHT

(b) MHT with CCM

Fig. 1. (a) and (b) show the difference between MHT with and without CCM during the tree generation process. Each circle represents a detection node in the tracking trees. Dummy nodes are shown in red dashed circles to represent potential missing detections.

- A novel continuous consistency model (CCM) to classify detections into four typical categories and describe the correlation consistency among detections;
- A method to restrain the exponential growth of MHT by exploiting CCM and thus significantly reducing the computational time;
- Removing the constraints on the number of dummy nodes and reducing the ID switch errors.

The rest of the paper is organized as follows. Related work is discussed in Sec.II. Our novel continuous consistency model is described in Sec.III. Optimization for multiple hypothesis tracking with CCM is presented in Sec.IV. Experiment results are shown in Sec.V followed by conclusion in Sec.VI.

2 Related Work

Tracking multiple targets has became one of the most popular topics in computer vision. It is a technique to locate targets in every single frame and recover trajectories through the whole video. There are generally two types of tracking approaches, online tracking and offline tracking. Online trackers [3]–[5] use past and current observations to generate trajectories. They provide strong real-time performance to fit requirements in real applications. However, early errors cannot be revised in these trackers. On the other hand, offline trackers [1], [6]–[8] consider all observations in a batch of frames or even the entire video. In this section, we will review some typical related work.

Tracking-by-detection is an acknowledged framework for multi-target tracking. It regards tracking as a data association problem. Most of the recent successful trackers are developed based on this method. By obtaining separate observations from the detector, the main task of tracking turns to building associations and constraints between detections. Towards this end, various approaches are proposed.

Zhang *et al.* [9] proposed a network flow based optimization method for tracking. It mapped data association problem into a cost-flow network with a non-overlaping constraint on trajectories and found the optimal solution by a min-cost flow algorithm. Later, Pirsiavash *et al.* [10] analyzed the number of tracks as well as their birth and death states and gave global solution with a greedy algorithm. Butt *et al.* [11] incorporated higher-order track smoothness constraints into tracking. Unlike previous methods, a node in their network represents a candidate pair of matching observations between consecutive frames. However, such a formulation cannot be solved by min-cost network flow algorithm. As a result, they proposed an iterative solution using Lagrangian relaxation. Chari *et al.* [12] added pairwise costs to the min-cost network flow framework and designed a convex relaxation solution to solve this NP-hard problem. Dehghan *et al.* [13] presented a new Target Identity-aware Network Flow (TINF) where the detection and data-association are performed simultaneously. They used structured learning method to learn a model for each target and to infer the best locations of all targets. To better cope with long term occlusions, McLaughlin *et al.* [14] added special edges to the tracking graph based on a motion model. These edges linked distant tracklets based on motion similarity. Later, Wang *et al.* [15] made it possible to track occluded objects by using the presence of other objects that contain them. But occlusion is not the only cause of detector failure, illumination or gesture changes can also cause detection failures.

In addition to these network flow based trackers, Milan *et al.* have made several works on tracking for years as follows. In [16], they formulated multi-target tracking as minimization of a continuous energy function and constructed an optimization scheme to find strong local minima of the energy. Later in [17], a discrete-continuous optimization problem was proposed to handles each aspect in its natural domain, such as target dynamics, mutual exclusion and track persistence. Data association was performed using discrete optimization while trajectory estimation is posed as a continuous fitting problem. In subsequent work, they also began to consider occlusion in the tracking problem. As a result, a conditional random field (CRF) was built in [18]. It concerned both the conflict between observations and the exclusion between trajectories. An expansion move-based MAP estimation scheme was proposed to solve the CRF problem. In order to reduce the impact of occlusion, they took superpixel information into account [19]. Every superpixel was assigned to a specific target or classified as background. However, the accuracy of superpixel segmentation and classification drops dramatically when the scene gets complicated.

More recently, the multi-cut based trackers have shown impressive results. Tang *et al.* [7], [20], [21] linked and clustered plausible detections jointly across space and time and thus stated the multi-target tracking as a minimum cost subgraph multi-cut problem. Although they employed a feasible optimization algorithm to solve the problem, it still suffered from its low efficiency.

Multiple Hypothesis Tracking (MHT) is another traditional method for tracking. MHT was first developed for sensor systems by Reid [22]. It measures the probabilities of each branches and unlikely hypotheses are eliminated while the trees are growing. Cox *et al.* [23] proposed an efficient implementation and suggested that it is possible to use MHT for visual tracking. Papageorgiou *et al.* [24] described the data association problem as a maximum weight independent set problem (MWISP) and further developed MHT for tracking. Later, Kim *et al.* [1] demonstrated that MHT can show competitive results by exploiting appearance models. In addition, Manafifard *et al.* [25] relied on particle swarm optimization (PSO) to account for nonlinear movements and occlusions in addition to appearance. The key of building a practical MHT-based tracker is to control the exponential growth of branches and to improve the efficiency and accuracy of pruning. We have done some work to improve the performance of MHT previously. In [26], we estimated the correlations between detections and improved the performance in distinguishing adjacent hypotheses. In [27], we built a fused association graph to use both detections and superpixels to enhance the robustness of MHT when detector fails. In addition, we proposed a tracking-by-tracklet framework to improve the efficiency of MHT in [2]. However, it acquires technical tuning on the length of tracklets and the size of the tracking window to make a balance between speed and accuracy.

Deep learning methods [28]–[31] have shown impressive results on single target tracking on both accuracy and efficiency. As for tracking multiple targets, these trackers cannot effectively deal with the identity switch problem due to the heavy and long-term occlusions. However, there are some other strategies of using deep learning. One commonly used approach is to describe the appearance of targets by using the features from Re-ID tasks [21], [32]. Compared with traditional feature descriptions like SIFT or HOG, these deep learning based features are more robust when calculating the similarity between targets. Another idea of using deep learning is to build an end-to-end tracking network. There are some attempts such as [33], [34]. They input the video and output trajectories directly. The

most severe shortcoming of these end-to-end trackers is the overfitting problem. The association relationship between targets is usually complicated and variable, so it is hard to design proper and enough data for training.

3 Continuous Consistency Model

3.1 Preliminary

Similar to most of the tracking-by-detection methods, our proposed tracker in this paper also uses detections (or called observations) as input. Detections of the same target are associated to a complete trajectory, and trajectories of different targets are identified by different labels. For a specified detection, it can be described as a 4-dimensional vector $d = (x, y, w, h)$, where x and y are the horizontal axis and vertical axis of the foot point (midpoint of bottom edge); w and h represent the width and the height of the detection. The trajectory of each target can be described as a set of detections from different frames $T_i = \{d_1, d_2, ..., d_t\}$, where d_t can be a detected observation from the detector or an estimated observation from the tracker in frame t.

3.2 Consistency Analysis

In a real-world scene, for each individual target, its trajectory should be complete and continuous. In addition, most of the time there is a fact that pedestrians will neither appear out of nowhere nor disappear suddenly in the scene (regardless of the extreme situations such as excessive concentration of obstacles, large changes in camera angle of view, low image resolution, etc.). This means that there is correlation consistency among the detections between adjacent frames due to their continuous trajectories. In this paper, we propose a continuous consistency model (CCM) to describe this correlation among detections by classifying detections.

For a given distance threshold, each detection could be associated to other detections in adjacent frames, or no detection meets the threshold. We consider four typical situations to introduce the CCM in detail as shown in Figure 2.

Figure 2(a) shows an ideal situation that a detection has both predecessor and successors. It means that d is a potential extension for the previous frame and it has candidate detections in the next frame. Figure 2(b) shows another situation that a detection only has successors. It happens when the target does not appear in the scene (out of the scene or occluded) in frame $t-1$ or its detections are missed until frame t. Figure 2(c) shows an opposite situation to (b). The target disappears in frame $t+1$ or the detector is failed. In addition, there is another situation shown in Figure 2(d) that a detection has neither predecessor nor successor. In this case, d could be a false detection or the detections in both adjacent frames are missed by the detector. Thus, CCM can be described in the following form.

$$CCM(d_t) = \begin{cases} D^* & |d_t, d_{t-1}| > 0, |d_t, d_{t+1}| > 0, \\ D^- & |d_t, d_{t-1}| > 0, |d_t, d_{t+1}| = 0, \\ D^+ & |d_t, d_{t-1}| = 0, |d_t, d_{t+1}| > 0, \\ D^0 & |d_t, d_{t-1}| = 0, |d_t, d_{t+1}| = 0. \end{cases} \quad (1)$$

In Eq. 1, operator $|d_i, d_j|$ means the number of connections between d_i and d_j when given a distance threshold. In this way, we use CCM to classify detections into four categories. Detections that have predecessors and at least one successor are classified into D^* (continuous detections); detections that only have predecessors are labeled as D^- (left continuous detections); detections that only have successors are labeled as D^+ (right continuous detections); detections have neither predecessor nor successor are regarded as D^0 (discontinuous detections).

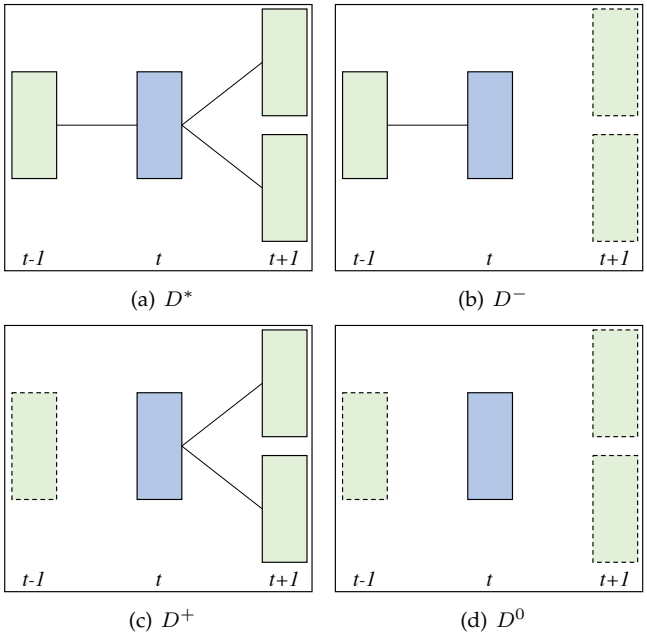


Fig. 2. (a), (b), (c) and (d) show four typical situations for detection d (in blue) in frame t. Green boxes present detections in adjacent frames ($t-1$ and $t+1$). Detections missed by detector are represented by dashed boxes. Detections are connected if the Euclidean distance of their foot points is less than the given threshold.

3.3 Maximum Weight Bipartite Graph Matching

To divide detections into four categories by CCM, we need to build the connections between detections in adjacent frames. For a given frame t, we can find a maximum matching between the weight bipartite graph of frame $t-1$ and t, and another maximum matching between frame t and $t+1$. Therefore, CCM can be regarded as a series of maximum matching of weight bipartite graph problems through frames.

For a given edge $e = \{d_1, d_2\}$, we learn the weight w_e by their distance and appearance features as follows.

$$\begin{aligned} f_1 &= \frac{1}{1 + \|d_1, d_2\|} \\ f_2 &= \frac{1}{2} + \frac{\|a_1, a_2\|}{2 \cdot a_1 \cdot a_2} \\ w_e &= f_1 \cdot f_2 \end{aligned} \quad (2)$$

where f_1 represents the Euclidean distance of the detections and f_2 is the cosine distance of their appearance features. We construct a 256-dimensional vector for each detection to describe its appearance.

Finding maximum weight bipartite graph matching is a traditional problem and can be effectively solved by Hungarian algorithm. We describe our matching algorithm

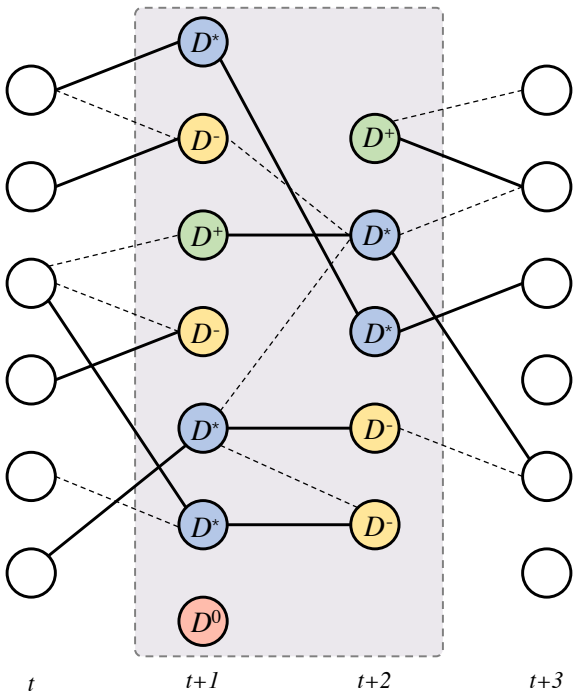


Fig. 3. Illustration of maximum weight bipartite graph matching. Four kinds of detections in frame $t+1$ and $t+2$ are shown in blue, yellow, green and red circles respectively.

Algorithm 1 Matching Algorithm

Input: weights of edges between frame $t-1$ and frame t
Output: Optimal matching

Step 1. Define matrix $M_{n1\times n2}$, where n_1 is the number of target in frame $t-1$ and n_2 is the number of target in frame t. The element $e_{i,j}$ in M is defined as $1-w_{i,j}$, where $w_{i,j}$ is the weight of edge $\{d_i, d_j\}$.

Step 2. Add dummy rows or columns with zeros to make M square.

Step 3. Every row subtracts its smallest element, so that there is at least one zero in each row.

Step 4. Every column subtracts its smallest element, so that there is at least one zero in each column.

Step 5. Cover all zeros with a minimum number of lines. If the number of lines equals the size of M, an optimal matching is found, go to Step 7.

Step 6. Find the smallest element that is not covered by the lines in Step 5. Subtract it from all uncovered elements, and add it to all elements at the intersections of the lines. Then, go to Step 5.

Step 7. Choose zeros in different rows and columns, so that the corresponding elements in the original matrix (M) is the optimal assignment.

TABLE 1

Attribute	Description
x_t, y_t, w_t, h_t	Detection's information of the node in frame t
c_t	Confidence of the detection in frame t
a_t	Appearance of the detection in frame t
c_m	Highest confidence of the detection in the hypothesis before frame t
a_m	Appearance of the detection with the highest confidence before frame t

as Alg. 1, and its time complexity is $O(n^3)$. According to the definition of CCM in Eq. 1, nodes that have matching between both previous and next frames are classified as D^*; nodes that only have matching between previous frame are D^-; nodes only match other nodes in the next frame are D^+; isolate nodes are defined as D^0. As shown in Figure 3, detections are divided into four categories according to the matching results.

4 Optimization for Multiple Hypothesis Tracking

4.1 MHT Overview

MHT is a breadth-first search algorithm. It solves tracking problem by generating multiple trees and evaluating each branch with a similarity score, and then selecting the most promising trajectories. The node in the track proposal is a detection from the detector or an estimated dummy node.

There are four main processes in the MHT, constructing, updating, scoring and conflict eliminating. First, new trees are constructed in each frame. The root node of the new tree is the detection in the frame, representing a new track proposal. Secondly, existing trees are expanded with the new coming detections in the frame. Meanwhile, trees are also extended with dummy nodes. Then, every leaf node is scored to evaluate the branch. However, as every detection should only represent one target, the nodes of the same detection in different trees may have a conflict. Finally, to address this problem, Maximum Weighted Independent Set (MWIS) algorithm is used to find the best set of the proposals [24]. The score of the proposal weights the edge in the MWIS problem, and only selected proposal are kept and updated in the next frame.

In this paper, we mainly focus on the first two parts, construction and updating, in which the scale of trees grows exponentially. We present a novel framework for MHT as shown in Figure 4 and how CCM makes contributions to controlling the scale of the tracking trees. The attributes of the nodes that we use in this paper are summarized in Table 1.

4.2 CCM for Tree Construction and Updating

The goal of MHT is to keep as much potential hypotheses as possible and make decisions later. However, due to the growing scale, traditional MHT-based trackers adopt sophisticated pruning heuristics to avoid the scale from being unacceptable. As a result, lots of hypotheses are pruned improperly.

Construction and scoring are two important processes that affect the scale of track trees. In addition to pruning

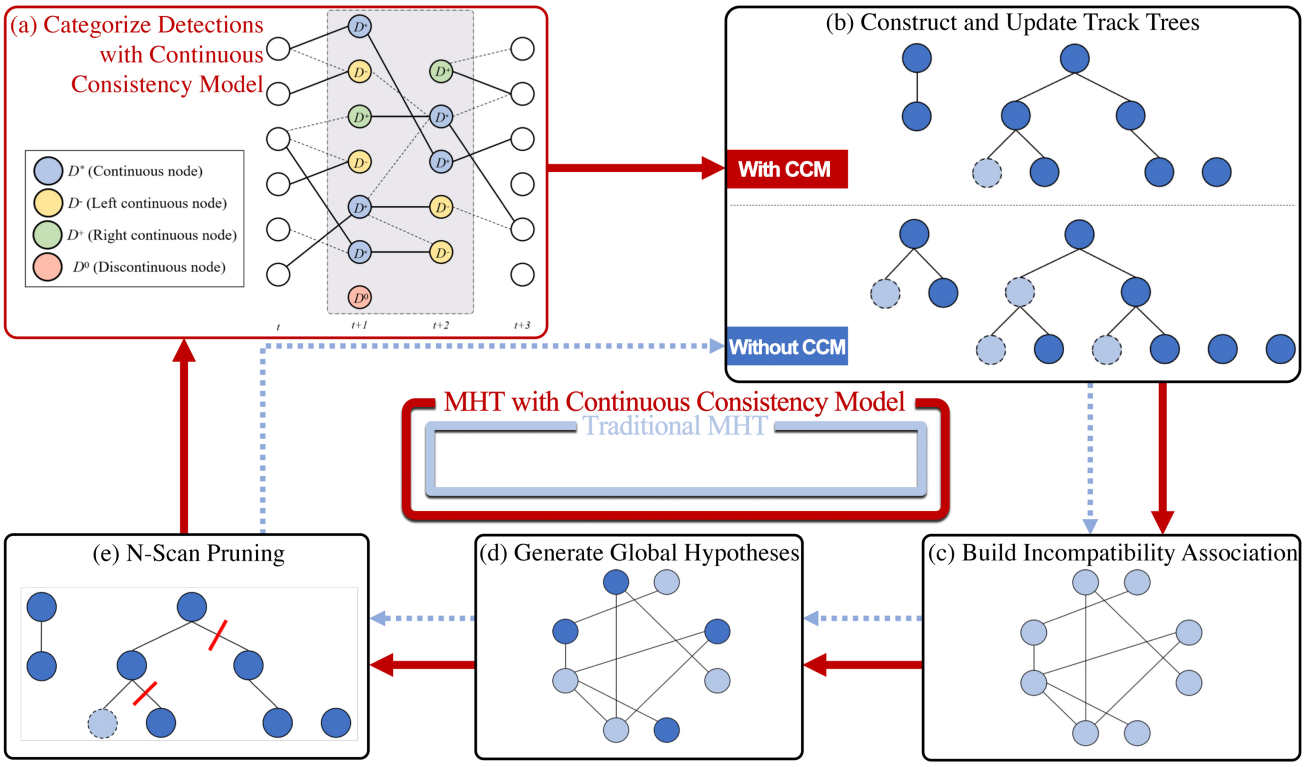


Fig. 4. Framework of MHT with continuous consistency model. Detections are categorized into four types before constructing and updating track trees.

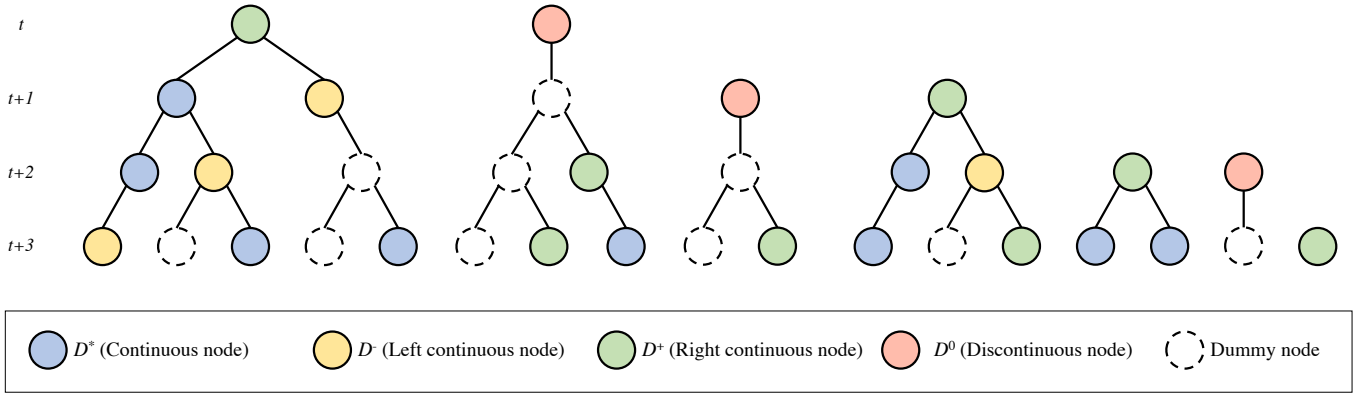


Fig. 5. Constructing and updating process of the track trees with CCM. As depicted in the legend, four kinds of detections and dummy nodes are shown. Only D^+ and D^0 are used as the root nodes, while only D^-, D^0 and dummy nodes are extended with dummy nodes.

strategies and parameters, a large number of dummy nodes also significantly lead to the rapid expansion of trees. Based on this aspect, we propose an approach to control the number of track trees and the dummy nodes by CCM and thus reduce the scale of MHT.

Unlike traditional approaches, we do not construct a new tree for every detection, only detections belonging to D^+ and D^0 are built as the root of the trees. According to the discussion of CCM, D^- and D^* detections are most likely extensions of other detections in the previous frame. Therefore, it is a reasonable decision to not construct new trees for them.

As shown in Figure 5, another difference from the traditional approaches is the updating strategy. Not all leaf nodes are extended with dummy nodes, only if a leaf detection belongs to D^- or D^0, or a leaf node is a dummy node. As defined by CCM, D^+ or D^* detection has great possibility to have a potential extension in the next frame. Extending D^+ or D^* detections with dummy nodes is more likely to introduce interference terms than generate possible hypotheses. In addition to dummy nodes, each leaf node is extended with new coming detections that satisfy the distance threshold. The distance between detections is defined as the Euclidean distance of their foot points as

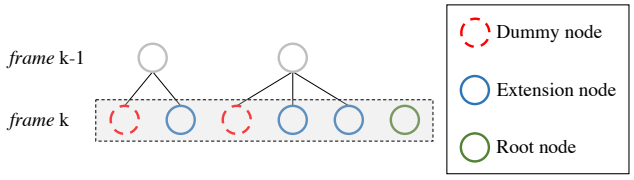


Fig. 6. Illustration of three kinds of nodes in frame k for the calculation of the scale of the track trees.

follows.

$$d_{1,2} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{3}$$

Dummy nodes share the same width, height and appearance features as their parent nodes. Their locations are estimated by linear interpolation. We have made consideration of not using more complex methods such as nonlinear regression or Bayesian estimation. The location of the pedestrian is decided by the motion of itself and the camera. The accuracy of the detectors has been greatly promoted in the recent past. In 5,919 frames of MOT17Det [35], there are only 7,599 false positives in SDP detector and 10,081 in FRCNN detector, compared with 42,308 in DPM proposed in 2010. Sudden changes of the location and the size of the detections hardly happen by using advanced detectors. As a result, using more complex methods does not provided a much more accuracy prediction. In addition, there is a fact that in scenes taken by moving cameras, the motion of the cameras is almost unpredictable. Based on these two points, as well as location prediction is not the focus of this paper, we decided to use a simple method for prediction to improve the efficiency of the tracker.

4.3 Scale Analysis

We now discuss the difference in the scale of the track trees with and without CCM theoretically. In MHT-based trackers, B_{th} is a tunable parameter to control the maximum number of tree branches. As it only works in few extreme situations, we do not consider its influence in the following analysis (regarding B_{th} as $+\infty$).

First, we discuss the scale of track trees in traditional MHT. For a giving frame, there are three kinds of nodes as shown in Figure 6. They are indicated as d_k, e_k and r_k respectively in Eq. 4. The total number of nodes in frame k is represented as n_k. N_k is the number of all nodes from frame 1 to k. They are calculated as follows.

$$\begin{aligned} n_k &= d_k + e_k + r_k \\ &= n_{k-1} + e_k + r_k \\ &= n_1 + \sum_{i=2}^{k} e_i + \sum_{i=2}^{k} r_i \\ \because n_1 &= r_1, e_1 = 0 \\ \therefore n_k &= \sum_{i=1}^{k} e_i + \sum_{i=1}^{k} r_i \\ \Rightarrow N_k &= \sum_{i=1}^{k} n_i = \sum_{m=1}^{k} \sum_{i=1}^{m} (e_i + r_i) \end{aligned} \tag{4}$$

Then we analyze the contribution of CCM for MHT. We use $n_k^{'}$ and $N_k^{'}$ to represent the number of nodes to avoid ambiguity. When using CCM for updating and constructing, only left continuous (D^-) and discontinuous detections (D^0) are extended with dummy nodes, and only right continuous (D^+) and discontinuous detections (D^0) are used as the root nodes. Compared with traditional MHT, the number of nodes is decreased according to the result of bipartite matching in CCM. We use p to represent the probability of finding a succeed for a node (D^+ and D^*). Obviously, it is the same probability of finding a preorder for a node (D^- and D^*) because the matching process between adjacent frames are done at the same time. Hence, $n_k^{'}$ and $N_k^{'}$ can be calculated as follows.

$$\begin{aligned} n_k^{'} &= d_k^{'} + e_k^{'} + r_k^{'} \\ &= n_{k-1}^{'} + e_k^{'} + r_k^{'} - p_k \cdot (n_{k-1}^{'} + e_k^{'} + r_k^{'}) \\ &= (1 - p_k) \cdot (n_{k-1}^{'} + e_k) + (1 - p_k) \cdot r_k \\ &= n_1^{'} + \sum_{i=2}^{k} (1 - p_k) \cdot e_i + \sum_{i=2}^{k} (1 - p_k) \cdot r_i \\ \because n_1^{'} &= r_1, e_1 = 0 \\ \therefore n_k^{'} &= \sum_{i=1}^{k} (1 - p_i) \cdot e_i + \sum_{i=1}^{k} (1 - p_i) \cdot r_i \\ \Rightarrow N_k^{'} &= \sum_{i=1}^{k} n_i^{'} = \sum_{m=1}^{k} \sum_{i=1}^{m} (1 - p_i) \cdot (e_i + r_i) \end{aligned} \tag{5}$$

Comparing N_k and $N_k^{'}$, the scale of track trees is decided by the matching rate of CCM which is significantly influenced by the detector. The quantitive comparison are presented in the experiment section.

4.4 Scoring

During construction and updating, we mark the node in each branch that has the highest confidence (provided by detectors). The score of the branch is recursively defined as follows.

$$\begin{aligned} S_0 &= 0 \\ S_t &= S_{t-1} + s_{mot} + s_{app} + s_{appM} \end{aligned} \tag{6}$$

where s_{mot} denotes the physical distance between detections, calculated as same as f_1 in Eq. 2. The latter two parts evaluate the appearance similarity locally and globally. s_{app} is the normalized cosine distance between a_t and its parent a_{t-1}, and s_{appM} is the normalized cosine distance from a_t to a_m.

5 Experiments

5.1 Datasets and Metrics

Our proposed tracker are evaluated on both MOT Challenge 2016 [35] and 2017. It is a widely used benchmark for multi-target tracking. There are 14 sequences (7 training, 7 test) with 11,235 frames in MOT 2016, and 42 sequences (21 training, 21 test) with 33,705 frames in MOT 2017. MOT 2017 consists of the same video as MOT 2016 but has different

sets of detections for each video by three detectors including DPM [36], FRCNN [37] and SDP [38].

All the detections used in the experiments are provided by MOT benchmark for a fair comparison.

We adopt the CLEAR MOT metrics [39] for quantitive evaluation. MOTA↑ (multiple object tracking accuracy) is a combination of FP↓ (false positives), FN↓ (false negatives) and IDS↓ (identity switches). IDF1↑ [40] is the ratio of correctly identified detections over the average number of ground truth and computed detections. MOTA and IDF1 are two important metrics to evaluate trackers. The former is primarily concerned with whether the targets are tracked, while the latter focuses on whether the targets are labeled with correct ID. When MOTA shows similar results, IDF1 is more capable to evaluate trackers on tracking targets consistently. In addition, MT↑ (mostly tracked, $> 80\%$), ML↓ (mostly lost, $< 20\%$), track fragmentations (FM)↓ and Hz↑ (processing speed, frames per second) are also reported. The indicator ↑ means the higher the better and ↓ means the lower the better.

5.2 Implementations

Considering that the elapsed time comparison is one of the major part in this paper, we run trackers under the same hardware configuration as follows, Intel Core i7-8700K@3.7GHz, 32GB DDR4-2400MHz, 1TB SATA3 HDD-10000rpm. Unless otherwise indicated, we use the same parameters in their paper for MHT_DAM [1] and TLMHT [2] for comparison. Specifically, N_{scan} pruning parameter N = 5, maximum number of tree branches B_{th} = 100 and distance threshold d_{th} = 12 for both Eq. 1 and Eq. 3. In addition, we extract convolutional neural network features from GoogLeNet [41] as the appearance features of the detections.

5.3 Effectiveness Analysis

One of the main goals of this paper is to promote the effectiveness of MHT by controlling its scale. In this section, we compared our proposed CMT tracker with other MHT-based trackers, including TLMHT [2], HAF [27] and MHT_DAM [1].

HAF and MHT_DAM are designed based on tracking-by-detection framework. Branches are extended node by node according to the detections in each frame. TLMHT is a tracking-by-tracklet tracker as it extends its branches by tracklets instead of a single detection. However, all these trackers are extended with dummy nodes to represent missing detections for every branch, and every detections (or tracklets) are constructed as new trees in the coming frame. In contrast, CMT tracker only extends dummy nodes to D^- and D^0, and chooses D^+ and D^0 as the root nodes.

TABLE 2
Number of Dummy Nodes on Different Detectors

Detector	Method	Dummy	Total	Ratio
DPM	Ours	415,664	742,257	56.0%
	TLMHT [2]	264,357	480,467	55.0%
	HAF [27]	2,311,775	4,444,972	52.0%
	MHT_DAM [1]	2,057,198	3,609,119	57.0%
FRCNN	Ours	169,448	305,858	55.4%
	TLMHT (len=3) [2]	134,471	249,112	54.0%
	HAF [27]	1,211,902	2,372,264	51.9%
	MHT_DAM [1]	1,040,795	1,763,691	59.0%
SDP	Ours	267,722	520,841	51.4%
	TLMHT (len=3) [2]	248,742	441,728	56.3%
	HAF [27]	2,033,271	3,597,202	56.5%
	MHT_DAM [1]	1,906,411	3,280,139	58.1%

To verify the effectiveness of CCM on controlling the scale of the track trees, we count the number of nodes during tracking with different detections. As shown in Table 2, the number of the dummy nodes accounts for about half of all nodes (Ratio column). CMT tracker dramatically reduces the number of nodes. There are only 20.6%, 17.3% and 15.9% nodes compared with MHT_DAM.

TABLE 3
Recall Rate of Different Detectors

Detector	TP	FN	TP+FN	Recall rate
DPM [36]	78,007	36,5578	114,564	68.1%
FRCNN [37]	88,601	25,963	114,564	77.3%
SDP [38]	95,699	18,865	114,564	83.5%

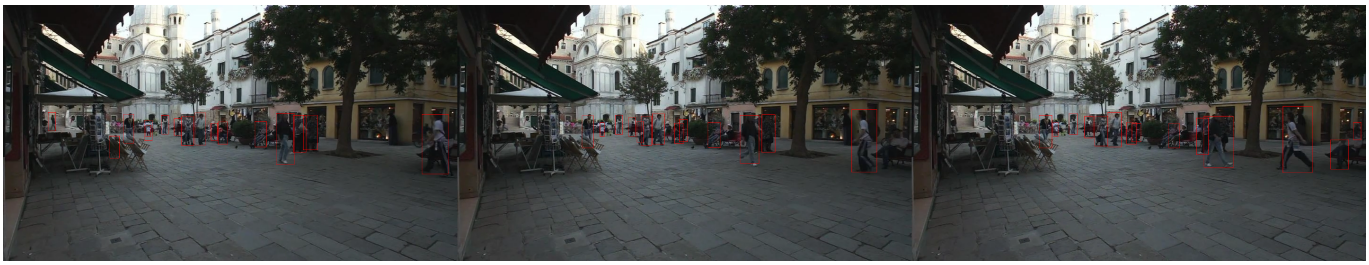
As discussed in Sec.4.3, the number of nodes in CMT tracker is mainly influenced by the performance of detector. We list the recall rate of different detector in Table 3. It shows that p in Eq. 5 is directly proportional to the recall rate. It means that higher recall rate of the detector results in more significant reduction on nodes by CMT tracker.

$$p \propto recall\ rate = \frac{TP}{TP + FN} \tag{7}$$

In addition to the fewer nodes, we achieve better tracking results on both MOT 2016 and MOT 2017 as shown in Table 4 and Table 5. Compared with MHT_DAM, by removing the constraints on the maximum number of consecutive dummy nodes, we have stronger ability to track targets under long-term occlusion where detections are missed. We reduce IDS by 128 on MOT 2016 and 956 on MOT 2017, therefore improve 8.2 and 8.1 on IDF1, 4.0 and 3.9 on MOTA respectively.

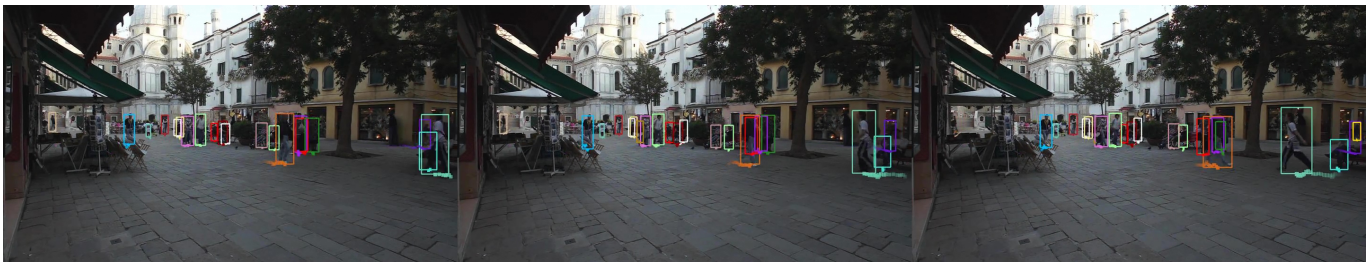
The continuity of the trajectories is an important constraint in tracking task, but traditional MHT methods do not consider the continuity relationship among detections between adjacent frames when constructing and expanding tracking trees. CCM is used to classify the detections into different categories and therefore make constraints for detections to describe the continuity of trajectories. As a result, we not only reduce the number of nodes in the tracking trees, but also suppress the generation of wrong branches at the same time.

Figure 7 shows an example of tracking long-term occlusion targets. Three persons (tagged with red, purple and green in Figure 7(e)) are occluded by another person (tagged

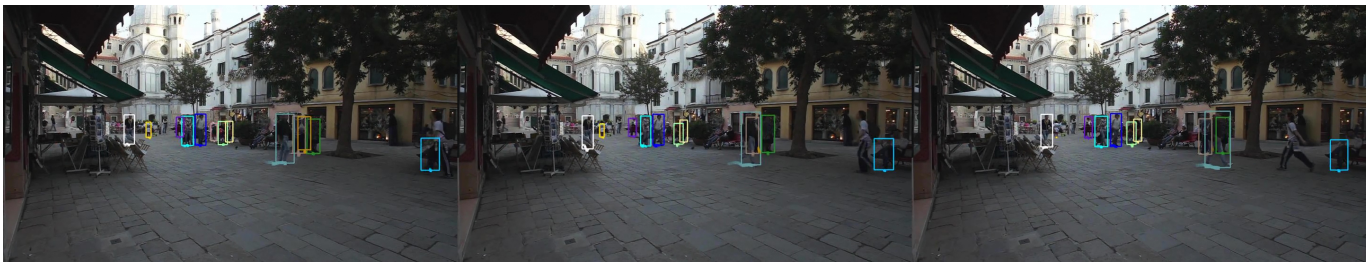


(a) Detections

(b) MHT_DAM [1]



(c) HAF [27]



(d) TLMHT [2]

(e) Ours

Fig. 7. Qualitative tracking results on MOT17-SDP-01 downloaded from MOT website. Three keyframes (frame 40, 50, 60) are shown in the figures. The public detections provided by MOT Challenge 2017 are shown in (a). Tracking results are presented in (b), (c), (d) and (e).

TABLE 4
Results on MOT 2016 Train

Method	IDF1↑	MOTA ↑	MT↑	ML↓	FP↓	FN↓	IDS↓	FM↓	Dummy	Total	Ratio
Ours	**57.4**	**44.3**	92	236	**4,941**	**56,348**	175	338	415,664	742,257	56.0%
TLMHT [2]	55.0	42.2	79	268	5,398	58,305	**157**	**308**	**264,356**	**480,467**	55.0%
HAF [27]	54.3	41.7	**93**	240	7,265	56,916	192	313	2,311,775	4,444,972	52.0%
MHT_DAM [1]	49.2	40.3	88	**230**	5,401	60,167	303	412	2,057,198	3,609,119	57.0%

TABLE 5
Results on MOT 2017 Train

Method	IDF1 ↑	MOTA ↑	MT↑	ML↓	FP↓	FN↓	IDS↓	FM↓	Dummy	Total	Ratio
Ours	**59.1**	**54.6**	**486**	568	12,266	**139,763**	759	1,203	1,035,871	2,031,119	51.0%
HAF [27]	56.8	52.2	463	**543**	11,854	146,242	1,214	1,474	7,644,139	14,415,778	53.0%
TLMHT [2]	56.4	51.2	338	714	**11,410**	152,443	**625**	**1,023**	**935,893**	**1,747,020**	53.6%
MHT_DAM [1]	51.0	50.7	422	566	11,743	150,667	1,715	1,520	7,186,727	13,559,862	53.0%

TABLE 6
MOT 2016 Sequences

Name	FPS	Platform	Method	IDF1↑	MOTA ↑	MT↑	ML↓	FP↓	FN↓	IDS↓	FM↓
MOT16-02	30	Static	**Ours**	**40.5**	**30.0**	8	**26**	605	**11,856**	31	43
			TLMHT [2]	40.2	26.3	**9**	35	**405**	12,724	**12**	**37**
			MHT_DAM [1]	38.1	25.2	8	29	654	12,644	48	55
MOT16-04	30	Static	**Ours**	63.5	51.5	14	26	2,299	20,715	55	103
			TLMHT [2]	**63.6**	**53.2**	**17**	**24**	**2,077**	**20,122**	**50**	**96**
			MHT_DAM [1]	52.1	45.6	14	29	2,543	23,214	116	147
MOT16-05	14	Moving	**Ours**	**54.0**	**40.5**	**20**	60	**250**	**3,791**	**13**	**37**
			TLMHT [2]	32.4	25.6	6	77	588	4,458	27	39
			MHT_DAM [1]	45.7	39.3	**20**	**54**	269	3,849	20	41
MOT16-09	30	Static	**Ours**	**64.8**	**62.7**	**10**	**4**	438	**1,487**	34	30
			TLMHT [2]	64.5	56.1	**10**	5	**374**	1,922	**14**	**24**
			MHT_DAM [1]	61.4	58.3	**10**	**4**	377	1,773	42	37
MOT16-10	30	Moving	**Ours**	**55.1**	**44.2**	**12**	26	**433**	**6,425**	20	63
			TLMHT [2]	52.1	37.0	8	31	651	7,096	**15**	**51**
			MHT_DAM [1]	49.9	41.0	9	**25**	585	6,642	39	61
MOT16-11	30	Moving	**Ours**	**67.7**	**54.6**	17	34	**473**	3,680	**9**	**16**
			TLMHT [2]	67.2	54.5	**19**	**30**	736	**3,409**	25	27
			MHT_DAM [1]	62.5	54.1	16	31	477	3,714	21	23
MOT16-13	25	Moving	**Ours**	**38.7**	**22.7**	**11**	60	**443**	8,394	**13**	46
			TLMHT [2]	33.8	20.0	10	66	567	8,574	14	**34**
			MHT_DAM [1]	35.2	**22.7**	**11**	**58**	496	**8,331**	17	48

with orange in Figure 7(e)) from frame 40 to 60. Their detections are missed by detectors due to the occlusion, so dummy nodes are expected to keep the hypotheses. However, MHT_DAM and TLMHT fail to track them due to the constraint on dummy nodes. To control the scale of trees, trajectories of more than 15 consecutive dummy nodes are discarded. In contrast, as the number of nodes is reduced by CCM dramatically, we do not have to worry the scale anymore and remove the constraint on dummy nodes. As a result, our method successfully tracks all of the three persons as shown in Figure 7(e).

5.4 Robustness Analysis

Compared with TLMHT, although CMT tracker has more nodes, it achieves better performance on both IDF1 (2.4 and 2.7) and MOTA (2.6 and 3.4) on MOT 2016 and MOT 2017, shown in Table 4 and Table 5. In this section, we further discuss the reason of the improvement.

MOT 2016 contains videos of different FPS (Frame Per Second) and platform (moving and static cameras). The performance of TLMHT is severely influenced by the quality of the tracklets. We make the detailed comparison among CMT, TLMHT and MHT_DAM as shown in Table 6.

For videos with high FPS, CMT and TLMHT have similar performance on IDF1; while for videos with low FPS, CMT remarkably outperforms TLMHT. On MOT16-05, a video with low FPS taken by moving camera, TLMHT performances even worse than MHT_DAM. It has lower IDF1 and MOTA than both CMT and MHT_DAM, while the MT is fewer than a half of them. In low FPS videos, targets are less coherent between adjacent frames, so it is difficult to generate reliable tracklets with expected length. The low quality tracklets directly pull down the performance of TLMHT. In contrast, CMT has better robustness on different types of videos.

5.5 Computational Time Analysis

Low computational efficiency and excessive time consumption have always been the fundamental problems for MHT methods. We make the following comparison experiments to analysis the computational efficiency of CMT tracker. TLMHT is a tracking-by-tracklet tracker, and the length of the tracklets greatly influences its performance and com-

putational efficiency. The time of generating tracklets is included and we set the length of the tracklet to 3 and 5 for comparison.

TABLE 7
Computational Time

Datasets	Method	time (s)
MOT 2016	Ours	658.0
	TLMHT (len=3) [2]	1579.3
	TLMHT (len=5) [2]	2043.4
	MHT_DAM [1]	6836.4
MOT 2017	Ours	2072.1
	TLMHT (len=3) [2]	5984.9
	TLMHT (len=5) [2]	7571.3
	MHT_DAM [1]	18442.1

Compared with MHT_DAM in Table 7, our method takes only 9.6% of the time to complete the tracking tasks in MOT 2016 and 11.2% at the time in MOT 2017. As for TLMHT, we also complete faster even by setting the length of tracklets to 3 for better efficiency.

5.6 Benchmark Comparison

Table 8 shows the results on MOT Challenge 2016. MOTA and IDF1 are two aggregative metrics to evaluate the performance of trackers. Our proposed CMT tracker takes the first place sorted by IDF1 score (56.6) and the third place sorted by MOTA (48.1). Compared with MHT_DAM, CMT16 outperforms it by 2.3 on MOTA and 10.5 on IDF1. IDS decreases from 590 to 381 which proves our tracker is more effective to keep possible hypotheses. As for TLMHT, our method achieves similar score on MOTA and promote IDF1 by 1.3.

In the more recent MOT Challenge 2017, tracking results are shown in Table 9. Compared with other MHT-based trackers, CMT gets similar score on MOTA while shows state-of-the-art performance by achieving the best score on IDF1, FN, IDS, FM and Hz.

The experimental results on both benchmarks show that our method is effective to achieve competitive tracking results while solving the efficiency problem of MHT.

6 Conclusion

In this paper, we propose the continuous consistency model (CCM) to categorize detections into four types, continuous, left continuous, right continuous and discontinuous detections. Unlike previous MHT tracking methods, we only extend the left continuous and discontinuous detections with dummy nodes, and choose right continuous and discontinuous detections as the root nodes. In this way, the exponential growth of the trees has been effectively controlled. In addition, we remove the constraint on dummy nodes to generate more complete trajectories when long-term occlusion happens. Our proposed CMT tracker shows dramatically improvement on the computational efficiency while achieving state-of-the-art results on MOT Challenge benchmark.

References

[1] C. Kim, F. Li, A. Ciptadi, and J. M. Rehg, "Multiple hypothesis tracking revisited," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4696–4704.

[2] H. Sheng, J. Chen, Y. Zhang, W. Ke, Z. Xiong, and J. Yu, "Iterative multiple hypothesis tracking with tracklet-level association," *IEEE Transactions on Circuits and Systems for Video Technology*, 2018.

[3] A. Sadeghian, A. Alahi, and S. Savarese, "Tracking the untrackable: Learning to track multiple cues with long-term dependencies," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 300–311.

[4] M. Yang, Y. Wu, and Y. Jia, "A hybrid data association framework for robust online multi-object tracking," *IEEE Transactions on Image Processing*, vol. 26, no. 12, pp. 5667–5679, 2017.

[5] Q. Chu, W. Ouyang, H. Li, X. Wang, B. Liu, and N. Yu, "Online multi-object tracking using cnn-based single object tracker with spatial-temporal attention mechanism," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4836–4845.

[6] A. Milan, K. Schindler, and S. Roth, "Multi-target tracking by discrete-continuous energy minimization," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 10, pp. 2054–2068, 2016.

[7] S. Tang, B. Andres, M. Andriluka, and B. Schiele, "Multi-person tracking by multicut and deep matching," in *European Conference on Computer Vision*. Springer, 2016, pp. 100–111.

[8] E. Levinkov, J. Uhrig, S. Tang, M. Omran, E. Insafutdinov, A. Kirillov, C. Rother, T. Brox, B. Schiele, and B. Andres, "Joint graph decomposition & node labeling: Problem, algorithms, applications," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[9] L. Zhang, Y. Li, and R. Nevatia, "Global data association for multi-object tracking using network flows," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

[10] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes, "Globally-optimal greedy algorithms for tracking a variable number of objects," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 1201–1208.

[11] A. A. Butt and R. T. Collins, "Multi-target tracking by lagrangian relaxation to min-cost network flow," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013, pp. 1846–1853.

[12] V. Chari, S. Lacoste-Julien, I. Laptev, and J. Sivic, "On pairwise costs for network flow multi-object tracking." in *CVPR*, vol. 20, 2015, p. 15.

[13] A. Dehghan, Y. Tian, P. H. Torr, and M. Shah, "Target identity-aware network flow for online multiple target tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1146–1154.

[14] N. McLaughlin, J. M. Del Rincon, and P. Miller, "Enhancing linear programming with motion modeling for multi-target tracking," in *Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on*. IEEE, 2015, pp. 71–77.

[15] X. Wang, E. Türetken, F. Fleuret, and P. Fua, "Tracking interacting objects using intertwined flows," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 11, pp. 2312–2326, 2016.

[16] A. Andriyenko and K. Schindler, "Multi-target tracking by continuous energy minimization," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 1265–1272.

[17] A. Andriyenko, K. Schindler, and S. Roth, "Discrete-continuous optimization for multi-target tracking," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 1926–1933.

[18] A. Milan, K. Schindler, and S. Roth, "Detection-and trajectory-level exclusion in multiple object tracking," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013, pp. 3682–3689.

[19] A. Milan, L. Leal-Taixé, K. Schindler, and I. Reid, "Joint tracking and segmentation of multiple targets," in *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*. IEEE, 2015, pp. 5397–5406.

[20] S. Tang, B. Andres, M. Andriluka, and B. Schiele, "Subgraph decomposition for multi-target tracking," in *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*. IEEE, 2015, pp. 5033–5041.

TABLE 8
Results on MOT Challenge 2016 Test

Method	IDF1↑	MOTA↑	MT↑	ML↓	FP↓	FN↓	IDS↓	FM↓	Hz↑
CMT16 (ours)	**59.2**	**49.8**	16.6%	43.6%	9,229	**81,882**	365	617	6.3
TLMHT [2]	55.3	48.7	15.7%	44.5%	6,632	86,504	413	642	4.8
NOMT [42]	53.3	46.4	18.3%	41.4%	9,753	87,565	**359**	**504**	2.6
LMP [21]	51.3	48.8	18.2%	40.1%	6,654	86,254	481	595	0.5
STAM16 [5]	50.0	46.0	14.6%	43.6%	6,895	91,117	473	1,422	0.2
GCRA [43]	48.6	48.2	12.9%	41.1%	5,104	88,586	821	1,117	2.8
EDMT [26]	47.9	45.3	17.0%	39.9%	11,122	87,890	639	946	1.8
NLLMPa [44]	47.3	47.6	17.0%	40.4%	5,844	89,093	629	768	**8.3**
FWT [45]	44.3	47.8	**19.1%**	38.2%	8,886	85,487	852	1,534	0.6
AMIR [3]	46.3	47.2	14.0%	41.6%	**2,681**	92,856	774	1,675	1.0
JMC [7]	46.3	46.3	15.5%	**39.7%**	6,373	90,914	657	1,114	0.8
MHT_DAM [1]	46.1	45.8	16.2%	43.2%	6,412	91,758	590	781	0.8
CDA_DDALv2 [46]	45.1	43.9	10.7%	44.4%	6,450	95,175	676	1,795	0.5
QuadMOT16 [47]	38.3	44.1	14.6%	44.9%	6,388	94,775	745	1,096	1.8

TABLE 9
Results on MOT Challenge 2017 Test

Method	IDF1↑	MOTA↑	MT↑	ML↓	FP↓	FN↓	IDS↓	FM↓	Hz↑
CMT (ours)	**60.7**	**51.8**	19.6%	42.8%	29,528	240,960	**1,217**	**2,008**	10.2
TLMHT [2]	56.5	50.6	17.6%	43.4%	22,213	255,030	1,407	2,079	2.6
DMAN [32]	55.7	48.2	19.3%	38.3%	26,218	263,608	2,194	5,378	0.3
HAF [27]	54.7	**51.8**	**23.4%**	37.9%	33,212	**236,772**	1,834	2,739	0.7
MOTDT17 [48]	52.7	50.9	17.5%	35.7%	24,069	250,768	2,474	5,317	18.3
MHT_bLSTM [31]	51.9	47.5	18.2%	41.7%	25,981	268,042	2,069	3,124	1.9
EDMT [26]	51.3	50.0	21.6%	36.3%	32,279	247,297	2,264	3,260	0.6
PHD_GSDL17 [49]	49.6	48.0	17.1%	**35.6%**	23,199	265,954	3,998	8,886	6.7
MHT_DAM [1]	47.2	50.7	20.8%	36.9%	22,875	252,889	2,314	2,865	0.9
IOU17 [50]	39.4	45.5	15.7%	40.5%	**19,993**	281,643	5,988	7,404	**1522.9**

[21] S. Tang, M. Andriluka, B. Andres, and B. Schiele, "Multiple people tracking by lifted multicut and person reidentification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3539–3548.

[22] D. Reid, "An algorithm for tracking multiple targets," *IEEE transactions on Automatic Control*, vol. 24, no. 6, pp. 843–854, 1979.

[23] I. J. Cox and S. L. Hingorani, "An efficient implementation of reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 18, no. 2, pp. 138–150, 1996.

[24] D. J. Papageorgiou and M. R. Salpukas, "The maximum weight independent set problem for data association in multiple hypothesis tracking," in *Optimization and Cooperative Control Strategies*. Springer, 2009, pp. 235–255.

[25] M. Manafifard, H. Ebadi, and H. A. Moghaddam, "Appearance-based multiple hypothesis tracking: Application to soccer broadcast videos analysis," *Signal Processing: Image Communication*, vol. 55, pp. 157–170, 2017.

[26] J. Chen, H. Sheng, Y. Zhang, and Z. Xiong, "Enhancing detection model for multiple hypothesis tracking," in *Conf. on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 2143–2152.

[27] H. Sheng, Y. Zhang, J. Chen, Z. Xiong, and J. Zhang, "Heterogeneous association graph fusion for target association in multiple object tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, 2018.

[28] X. Lan, S. Zhang, P. C. Yuen, and R. Chellappa, "Learning common and feature-specific patterns: a novel multiple-sparse-representation-based tracker," *IEEE Transactions on Image Processing*, vol. 27, no. 4, pp. 2022–2037, 2017.

[29] Y. Lu, C. Lu, and C.-K. Tang, "Online video object detection using association lstm," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2344–2352.

[30] L. Qu, K. Liu, B. Yao, J. Tang, and W. Zhang, "Real-time visual tracking with elm augmented adaptive correlation filter," *Pattern Recognition Letters*, 2018.

[31] C. Kim, F. Li, and J. M. Rehg, "Multi-object tracking with neural gating using bilinear lstm," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 200–215.

[32] J. Zhu, H. Yang, N. Liu, M. Kim, W. Zhang, and M.-H. Yang, "Online multi-object tracking with dual matching attention networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 366–382.

[33] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Detect to track and track to detect," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3038–3046.

[34] K. Kang, H. Li, J. Yan, X. Zeng, B. Yang, T. Xiao, C. Zhang, Z. Wang, R. Wang, X. Wang *et al.*, "T-cnn: Tubelets with convolutional neural networks for object detection from videos," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 10, pp. 2896–2907, 2018.

[35] A. Milan, L. Leal-Taixé, I. D. Reid, S. Roth, and K. Schindler, "MOT16: A benchmark for multi-object tracking," *CoRR*, vol. abs/1603.00831, 2016. [Online]. Available: http://arxiv.org/abs/1603.00831

[36] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[37] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[38] F. Yang, W. Choi, and Y. Lin, "Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2129–2137.

[39] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: the clear mot metrics," *EURASIP Journal on Image and Video Processing*, vol. 2008, no. 1, pp. 1–10, 2008.

[40] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," in *European Conference on Computer Vision*. Springer, 2016, pp. 17–35.

[41] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[42] W. Choi, "Near-online multi-target tracking with aggregated local flow descriptor," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3029–3037.

[43] C. Ma, C. Yang, F. Yang, Y. Zhuang, Z. Zhang, H. Jia, and X. Xie,

"Trajectory factory: Tracklet cleaving and re-connection by deep siamese bi-gru for multiple object tracking," in *2018 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2018, pp. 1–6.

[44] E. Levinkov, J. Uhrig, S. Tang, M. Omran, E. Insafutdinov, A. Kirillov, C. Rother, T. Brox, B. Schiele, and B. Andres, "Joint graph decomposition and node labeling: Problem, algorithms, applications," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[45] R. Henschel, L. Leal-Taixé, D. Cremers, and B. Rosenhahn, "Fusion of head and full-body detectors for multi-object tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 1428–1437.

[46] S.-H. Bae and K.-J. Yoon, "Confidence-based data association and discriminative deep appearance learning for robust online multi-object tracking," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 3, pp. 595–610, 2018.

[47] J. Son, M. Baek, M. Cho, and B. Han, "Multi-object tracking with quadruplet convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[48] C. Long, A. Haizhou, Z. Zijie, and S. Chong, "Real-time multiple people tracking with deeply learned candidate selection and person re-identification," in *Proc. IEEE Conf. Multimedia Expo*, 2018, pp. 1–6.

[49] Z. Fu, P. Feng, F. Angelini, J. Chambers, and S. M. Naqvi, "Particle phd filter based multiple human tracking using online group-structured dictionary learning," *IEEE Access*, vol. 6, pp. 14 764–14 778, 2018.

[50] E. Bochinski, V. Eiselein, and T. Sikora, "High-speed tracking-by-detection without using image information," in *Advanced Video and Signal Based Surveillance (AVSS), 2017 14th IEEE International Conference on*. IEEE, 2017, pp. 1–6.